LISTING OF THE CLAIMS

1. (Original) A method comprising:

establishing communication between a plurality of non-Java-based server nodes of a first instance and a plurality of Java-based server nodes of a second instance via an intermediate server;

generating a packet to be transmitted from one of the non-Java-based server nodes to one of the Java-based server nodes;

specifying in a header of the packet an address of a destination Java-based server node and information that indicates that the packet is generated by one of the non-Java-based server nodes;

forwarding the packet to the intermediate server from the one of the non-Java-based server nodes; and

forwarding the packet to the destination Java-based server node from the intermediate server based on the address provided in the header of the packet.

2. (Original) The method of claim 1, further comprising:

generating a second packet to be transmitted from one of the Java-based server nodes to one of the non-Java-based server nodes;

specifying in a header of the second packet an address of a destination non-Java-based server node and information that indicates that the packet is generated by one of the Java-based server nodes;

forwarding the second packet to the intermediate server from the one of the Java-based server nodes; and

forwarding the second packet to the destination non-Java-based server node from the intermediate server based on the address provided in the header of the second packet.

3. (Original) The method of claim 2, further comprising:

maintaining a list of services performed by the non-Java-based server nodes; and

sending notification of a status of each of the listed services to the non-Java-based server nodes in the first instance.

4. (Original) The method of claim 3, further comprising:

maintaining a list of services performed by the Java-based server nodes; and

sending notification of a status of each of the listed services to the Java-based server nodes in the second instance.

5. (Original) The method of claim 4, wherein the maintaining a list of services is accomplished by the intermediate server and the sending notification of a status of each of the listed services is accomplished by the intermediate server.

6. (Original) The method of claim 1, further comprising:

implementing Java 2 Platform Enterprise Edition (J2EE) applications in the Java-based server nodes.

7. (Currently Amended) A system comprising:

a first instance including a plurality of non-Java-based server nodes, each of the non-Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from one of the non-Java-based server nodes;

a second instance including a plurality of Java-based server nodes, each of the Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from one of the Java-based server nodes; and

a message server coupled between the first and second instances to establish communication there_between.

8. (Original) The system of claim 7, wherein each of the instances further comprises a dispatcher to distribute client requests to the server nodes of the respective instance.

9. (Original) The system of claim 7, wherein the message server is to route message packets between the non-Java-based server nodes of the first instance and the Java-based server nodes of the second instance.

10. (Original) The system of claim 7, wherein the message server is to assign a service identification associated with each type of services executed on the server nodes.

11. (Original) The system of claim 10, wherein the message server includes a service repository to maintain a list of the assigned service identification and corresponding service names.

12. (Original) The system of claim 7, wherein the message server further comprises:

a first repository to maintain a list of services currently being executed on the non-Java-based instances; and

a second repository to maintain a list of services currently being executed on the Java-based instances.

13. (Original) The system of claim 7, wherein the message server is to maintain a list of services performed by the instances and a status corresponding to each of the listed services, and to send notification of the status of the listed services to the instances.

14. (Original) The system of claim 7, wherein the Java-based instances are capable of implementing Java 2 Platform Enterprise Edition (J2EE) applications.

15. (Original) A message server comprising:

a first communication interface to establish communication with a plurality of non-Java-based server nodes;

a second communication interface to establish communication with a plurality of Java-based server nodes; and

a controller to transfer packets between the non-Java-based server nodes and the Java-based server nodes.

16. (Original) The message server of claim 15, wherein the controller is to assign a service identification associated with each type of services executed on the server nodes.

17. (Original) The message server of claim 16, further comprising:

a service repository maintain a list of the assigned service identification and corresponding service names.

18. (Original) The message server of claim 15, further comprising:

a first repository to maintain a list of services currently being executed on the non-Java-based server nodes; and

a second repository to maintain a list of services currently being executed on the Java-based server nodes.

19. (Original) The message server of claim 17, wherein the controller is to send notification of a status of each of the services listed in the first repository to the non-Java-based server nodes.

20. (Original) The message server of claim 17, wherein the controller is to send notification of a status of each of the services listed in the second repository to the Java-based server nodes.

21. (Currently Amended) A machine-readable medium that provides instructions, which when executed by a processor cause the processor to perform operations comprising:

establishing communication with a plurality of non-Java-based server nodes;

establishing communication with a plurality of Java-based server nodes; and

~~transfering~~transferring packets between the non-Java-based server nodes and the Java-based server nodes.

22. (Original) The machine-readable medium of claim 21, wherein the operations performed by the processor further comprise:

assigning a service identification associated with each type of services executed on the server nodes; and

maintaining a list of the assigned service identification and corresponding service names.

23. (Original) The machine-readable medium of claim 21, wherein the operations performed by the processor further comprise:

maintaining a list of services executed on the non-Java-based server nodes in a first repository; and

maintaining a list of services executed on the Java-based server nodes in a second repository.

24. (Original) The machine-readable medium of claim 23, wherein the operations performed by the processor further comprise:

sending notification of a status of each of the services listed in the first repository to the non-Java-based server nodes; and

sending notification of a status of each of the services listed in the second repository to the Java-based server nodes.

25. (Original) A system comprising:

means for generating a packet such that a header of the packet specifies an address of a destination Java-based server node and that the packet originated from one of non-Java-based server nodes;

means for forwarding the packet to intermediate communication means from the one of the non-Java-based server nodes; and

means for forwarding the packet to the destination Java-based server node from the intermediate communication means based on the destination address provided in the header of the packet.

26. (Original) The system of claim 25, further comprising:

means for generating a second packet such that a header of the second packet specifies an address of a destination non-Java-based server node and that the second packet originated from one of Java-based server nodes;

means for forwarding the second packet to the intermediate communication means from the one of the Java-based server nodes; and

means for forwarding the second packet to the destination non-Java-based server node from the intermediate communication means based on the destination address provided in the header of the second packet.

27. (Original) The system of claim 26, wherein the intermediate communication means further comprises:

means for maintaining a list of services performed by the Java-based server nodes; and

means for sending notification of a status of each of the listed services to the Java-based server nodes.

28. (Original) The system of claim 27, wherein the intermediate communication means further comprises:

means for maintaining a list of services performed by the non-Java-based server nodes; and

means for sending notification of a status of each of the listed services to the non-Java-based server nodes.

29. (Original) The system of claim 25, wherein the intermediate communication means further comprises:

means for establishing communication with a plurality of non-Java-based server nodes;

means for establishing communication with a plurality of Java-based server nodes; and

means for transferring packets between the non-Java-based server nodes and the Java-based server nodes.

30. (Original) The system of claim 25, wherein the intermediate communication means further comprises:

means for assigning a service identification associated with each type of services executed on the server nodes; and

means for maintaining a list of the assigned service identification and corresponding service names.